



Sources:

1: <https://www.taniarascia.com/getting-started-with-grunt-and-sass/>

2: <https://github.com/nDmitry/grunt-postcss>

3: <https://www.grafikart.fr/tutoriels/>

Automatiseur de tâches

Grunt

Ed 1.0 -
19 mars 2020

Kesako

- Dans un environnement JavaScript, comment réduire le nombre de tâches comme la compilation SCSS, CSS, ou la réduction de la taille du code JS
- 2 outils appelés: Task Runners JS basé sur node.js
- Grunt (2012) est basé sur la configuration d'un script (JSON)
% Gulp (2013) qui demande plus de programmation (JavaScript) avant la configuration
- Grunt a un avantage pour de petits projets
- Gestionnaire de paquets: npm

Installation

- A la racine de votre projet web
- Installer **Node.js** pour utiliser son package manager « npm » - equivalent du plugin pour wp
- **%npm install** => nouveau dossier node_modules
- La commande **%npm init** régénère un fichier package.json (details)
- Installer grunt **%npm install -g grunt-cli** (cli: command line interface)

Objectifs

- Réduire le nombre de fichier appelé par index.html (augmenter la performance du site), objectif : 1 seul fichier css et un seul fichier js
- Minimiser la taille des fichiers js (augmenter la performance du site)
- Automatiser toutes ces actions lors qu'un changement intervient dans les fichiers sources

Projet

1

Structure du projet
(Données entrées)
-base.scss
Style.scss
Script.js
Gruntfile.js

```
▼ module.exports = function (grunt) {  
  pkg: grunt.file.readJSON('package.json'),  
  // Configuration de Grunt  
  ▼ grunt.initConfig({  
    ▶ sass: {dist: { options: { style:...}},  
    ▶ postcss: { // Begin Post CSS Plugin ...},  
    ▶ cssmin: { // Begin CSS Minify Plugi...},  
    ▶ uglify: { // Begin JS Uglify Plugin...},  
    ▶ watch: { // Compile everything int...},  
  });  
  
  grunt.loadNpmTasks('grunt-contrib-sass')  
  grunt.loadNpmTasks('grunt-postcss')  
  grunt.loadNpmTasks('grunt-contrib-cssmin')  
  grunt.loadNpmTasks('grunt-contrib-uglify')  
  grunt.loadNpmTasks('grunt-contrib-watch')  
  
  // Register Grunt tasks  
  grunt.registerTask('default', ['watch'])  
};
```

```
'sass: {  
  dist: {  
    options: {  
      style: "expanded",  
    },  
    files: [{  
      expand: true,  
      cwd: 'sass',  
      src: ['**/*.scss'],  
      dest: 'css',  
      ext: '.css',  
    }],  
  },  
},
```

```
watch: {  
  // Compile everything into one task with Watch Plugin  
  css: {  
    files: '**/*.scss',  
    tasks: ['sass', 'postcss', 'cssmin'],  
  },  
  js: {  
    files: '**/*.js',  
    tasks: ['uglify'],  
  },  
},
```


Projet

2

Sous la racine du projet
Lancer grunt -> watch attend
toute modification d'un fichier
scss, css ou js pour lancer
les scripts

3

Modification du _base.scss
\$header-size passe de 3em
à 2em

```
x gruntfile.js x style.scss x _base.scss
1  /* CSS Document */
2
3  // Variables
4
5  $header-size: 2em;
6  $font-color: #343434;
7  $content-width: 1200px;
```

4

Watch détecte
et exécute les tâches
ad hoc

5

Résultat final



```
[imac-de-didier:sass-grunt didierflury$ pwd
/Applications/MAMP/htdocs/sass-grunt
[imac-de-didier:sass-grunt didierflury$ grunt
```

```
Replace Autoprefixer browsers option to Browserslist config.
Use browserslist key in package.json or .browserslistrc file.
```

```
Using browsers option can cause errors. Browserslist config
can be used for Babel, Autoprefixer, postcss-normalize and other tools.
```

```
If you really need to use option, rename it to overrideBrowserslist.
```

```
Learn more at:
https://github.com/browserslist/browserslist#readme
https://twitter.com/browserslist
```

```
Running "watch" task
Waiting...
>> File "sass/_base.scss" changed.
```

```
Replace Autoprefixer browsers option to Browserslist config.
Use browserslist key in package.json or .browserslistrc file.
```

```
Using browsers option can cause errors. Browserslist config
can be used for Babel, Autoprefixer, postcss-normalize and other tools.
```

```
If you really need to use option, rename it to overrideBrowserslist.
```

```
Learn more at:
https://github.com/browserslist/browserslist#readme
https://twitter.com/browserslist
```

```
Running "sass:dist" (sass) task
```

```
Running "postcss:dist" (postcss) task
>> 1 processed stylesheet created.
>> 1 sourcemap created.
```

```
Running "cssmin:target" (cssmin) task
>> 1 file created. 163 B → 125 B
```

```
Done.
Completed in 2.085s at Thu Mar 19 2020 12:55:27 GMT+0100 (GMT+01:00) - Waiting..
```

Concaténation

- **Concaténation** de fichiers JS séparés dans le projet: ex jquery.js, carousel.js, global.js, etc. Installer le package **%npm install grunt-contrib-concat --save-dev**
- **--save** création d'une dépendance dans le fichier package.jason - idem pour **--save-dev** pour la phase de dev
- Pour pouvoir utiliser ce package, il faut configurer **Grunt** via un fichier un créer **Gruntfile.js**

Concaténation

Gruntfile.fs

```
module.exports = function(grunt) {

// 1. Toutes les configurations vont ici:
grunt.initConfig({
  pkg: grunt.file.readJSON('package.json'),

  concat: {
    // 2. la configuration pour la concaténation va ici.
  }

});

// 3. Nous disons à Grunt que nous voulons utiliser ce plug-in.
grunt.loadNpmTasks('grunt-contrib-concat');

// 4. Nous disons à Grunt quoi faire lorsque nous tapons "grunt"
dans la console.
grunt.registerTask('default', ['concat']);

};
```

```
concat: {
  dist: {
    src: [
      'js/libs/*.js', // tous les JS dans libs
      'js/global.js' // ce fichier là
    ],
    dest: 'js/build/production.js'
  }
}
```

Attention à l'ordre des fichiers sans src

Dist: c'est le nom de cette concaténation appelée Dist comme distribution. On peut en avoir plusieurs.

D'autres plugins

... 6000

- Minifier Javascript : grunt-contrib-uglify
- Optimiser des images : grunt-contrib-imagemin

```
uglify: {  
  build: {  
    src: 'js/build/production.js',  
    dest: 'js/build/production.min.js'  
  }  
}
```

```
imagemin: {  
  dynamic: {  
    files: [{  
      expand: true,  
      cwd: 'images/',  
      src: ['**/*.{png,jpg,gif}'],  
      dest: 'images/build/'  
    }]  
  }  
}
```

- Automatisation : grunt-contrib-watch

```
watch: {  
  scripts: {  
    files: ['js/*.js'],  
    tasks: ['concat', 'uglify'],  
    options: {  
      spawn: false,  
    },  
  },  
}
```

Préprocessing css

- Sass : grunt-contrib-sass , il fait aussi la concaténation et la réduction de code
- Automatisation avec watch auquel il faudra ajouter une nouvelle tâche

```
sass: {  
  dist: {  
    options: {  
      style: 'compressed'  
    },  
    files: {  
      'css/build/global.css': 'css/global.scss'  
    }  
  }  
}
```

```
css: {  
  files: ['css/*.scss'],  
  tasks: ['sass'],  
  options: {  
    spawn: false,  
  }  
}
```

Et encore

- Avant de réduire le code js, il faut s'assurer de sa conformité aux règles d'écriture, il faut donc utiliser le plugin grunt-contrib-jshint **contrib** indique que c'est l'équipe grunt qui a développé ce plugin.
- Si on veut scanner tous les fichiers js dans tous les sous dossiers du dossier lib par ex: **lib/**/*.js**
- Si on veut exclure par ex un fichier min.js d'un dossier js d'une opération : **!js/min.js**

... et encore !

```
watch: {
  js: {
    files: ['js/*.js', '!js/min.js'],
    tasks: ['jshint', 'uglify'],
    options: { spawn: false }
  },
  css: {
    files: ['css/*.css', '!css/min.css'],
    tasks: ['cssmin'],
    options: { spawn: false }
  }
}
```

- Php code: tester si le code est correct (PSR ou PSR2 compliant) installer grunt-phpcs (via PEAR / Home-brew install)
- Js code : jshint
- Compresser css : cssmin - attention à l'ordre des fichiers.
- Load-grunt-tasks - il charge automatiquement les bon plugins - `require('load-grunt-tasks')(grunt);` \$ npm install --save-dev load-grunt-tasks
- Watch css et js en évitant certains fichiers minimisés
- Livereload
- Text-replace (ex remplacer les chemin des images qui se trouvaient dans « img/ » par le nouveau dossier «img/min/» si on a utiliser l'optimiser d'images.

```
// Gruntfile.js
module.exports = grunt => {
  // Load all grunt tasks matching the ['grunt-*',
  require('load-grunt-tasks')(grunt);

  grunt.initConfig({});
  grunt.registerTask('default', []);
};
```

Attention

- Uglify - angular - mantle - renommer les variables